Chapter 9

Finite element method

9.1 An introduction to the Finite Element Method

- Finite difference (FD) method is an approximation to the differential equation.
- Finite element method (FEM) is an approximation to its solution.
- FD methods are usually based on the assumption of regular domains eg line in 1-D, rectangle in 2-D with regular elements
- FEM is better for irregular regions as the domain can be partitioned into any simple subregion such as triangles or rectangles in 2-D or bricks and tetrahedra in 3-D. Figure 9.1 shows a finite element mesh with triangles for an irregular domain.

Example: Solving Poisson's equation in 1-D using FEM

$$-U_{xx} = q, \quad 0 \leq x \leq L \tag{9.1}$$

We consider Dirichlet boundary conditions: $U(0) = U(L) = 0$. A weak solution of (9.1) considers the variational form of (9.1):

$$\int_0^L U_{xx}(x)\phi(x)dx + \int_0^L q(x)\phi(x) = 0, \tag{9.2}$$

where $\phi(x)$ satisfy the boundary conditions: $\phi(0) = \phi(L) = 0$.

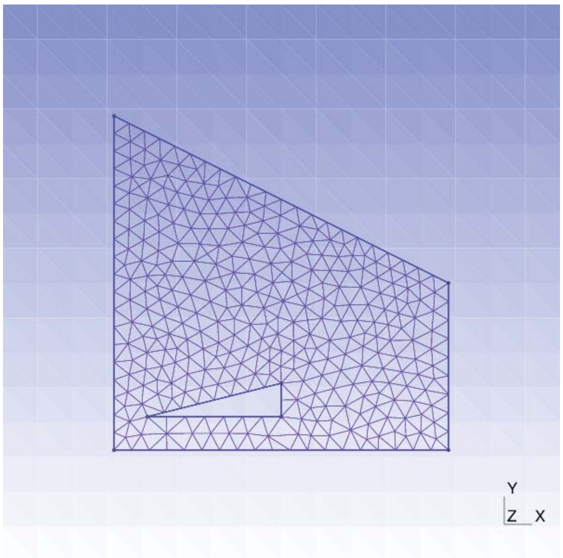


Figure 9.1: FEM mesh with triangles

We can integrate the first term by parts:

$$\begin{aligned}\int_0^L U_{xx}(x)\phi(x)dx &= U_x(x)\phi(x)]_{x=0}^{x=L} - \int_0^L U_x(x)\phi_x(x)dx \\ &= -\int_0^L U_x(x)\phi_x(x)dx\end{aligned}$$

using $\phi(0) = \phi(L) = 0$.

Then (9.2) becomes:

$$\int_0^L U_x(x)\phi_x(x)dx = \int_0^L q(x)\phi(x)dx \tag{9.3}$$

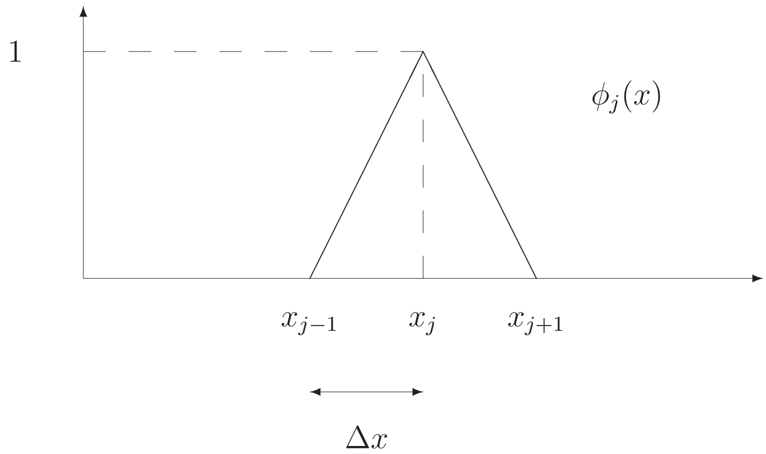
Equation (9.3) holds for all functions $\phi(x)$ which are piece-wise continous and satisfy the bc: $\phi(0) = \phi(L) = 0$.

To solve equation (9.3) using the FEM we again introduce a mesh (as in FD) on the interval $[0, L]$ with mesh points $x_j = j\Delta x$, $j = 0, \ldots, n+1$ where $\Delta x = \frac{L}{n+1}$. To complete the discretisation we must choose a basis for $\phi(x)$. The most common basis chosen for $\phi(x)$ are the "hat" functions, $\phi_j(x)$. We solve (9.3) using these:

$$\phi(x) = \sum_{j=1}^{n} a_j \phi_j(x)$$

where

$$\phi_j(x) = \begin{cases} 0, & \text{for } 0 \le x \le x_{j-1} \\ \frac{1}{\Delta x}(x - x_{j-1}), & \text{for } x_{j-1} \le x \le x_j \\ 1 - \frac{1}{\Delta x}(x - x_j), & \text{for } x_j \le x \le x_{j+1} \\ 0, & \text{for } x \ge x_{j+1} \end{cases}$$



with this construction: $\phi_j(x_i) = \delta_{ij}$ and:

$$\phi_j'(x) = \frac{\partial \phi_j}{\partial x} = \begin{cases} 0, & \text{for } 0 < x < x_{j-1} \\ \frac{1}{\Delta x}, & \text{for } x_{j-1} < x < x_j \\ -\frac{1}{\Delta x}, & \text{for } x_j < x < x_{j+1} \\ 0, & \text{for } x > x_{j+1} \end{cases}$$

We let $\phi(x) = \sum_{j=1}^{n} a_j \phi_j(x)$ and $\phi(x_i) = a_i$ for $i = 1, \ldots, n$, and $\phi(0) = \phi_1(0) = 0$ and $\phi(L) = \phi_n(L) = 0$ so that $\phi(x)$ satisfies boundary conditions.

The hat functions are advantageous as a basis as they are nearly "orthonormal", ie. $\int_0^L \phi_j(x)\phi_k(x)dx = 0$ when $|j-k| > 1$.

Using FEM we seek an **approximate** solution to (9.3) which is satisfied for all basis functions, $\phi_i(x)$, for $i = 1, \ldots, n$:

$$\int_0^L U_x(x)\phi_x(x)dx = \int_0^L q(x)\phi(x)dx$$

and require that 9.3 be satisfied for $\phi = \phi_i, \ i = 1, \ldots, n$. We also expand the solution $U(x)$ using the hat functions ϕ_i as a basis:

$$U(x) \approx U_h(x) = \sum_{j=1}^{n} b_j \phi_j(x)$$

This simplifies equation 9.3 and we solve for $\phi = \phi_i,\ i = 1, \ldots, n$: ie.

$$\int_0^L U_h'(x)\phi_i'(x)dx = \int_0^L q(x)\phi_i(x)dx, \quad \text{for } i = 1, \ldots, n$$

where $f'(x) = \frac{\partial f}{\partial x}$.

$$\begin{aligned} LHS &= \int_0^L U_h'(x)\phi_i'(x)dx \\ &= \int_0^L \sum_{j=1}^n b_j \phi_j'(x)\phi_i'(x)dx \\ &= \sum_{j=1}^n C_{i,j} b_j \end{aligned}$$

where $C_{i,j} = \int_0^L \phi_j'(x)\phi_i'(x)dx$. $C_{i,j}$ is known as the stiffness matrix in mechanics.

To find the coefficients b_j which define our solution $U(x)$ we must solve n linear equations:

$$LHS = \sum_{j=1}^n C_{i,j} b_j = RHS = \int_0^L q(x)\phi_i(x)dx = q_i \tag{9.4}$$

for $i = 1, \ldots, n$ with $q_i = \int_0^L q(x)\phi_i(x)dx)$.

We approximate the solution by expanding in the basis of "hat" functions: $U(x) \approx \sum_{j=1}^n b_j \phi_j(x)$. Thus we only need to know the coefficients b_j to define our solution $U(x)$ and FEM solves the following equation for vector $\vec{b} = (b_1, \ldots, b_n)$:

$$\begin{aligned} \sum_{j=1}^n b_j \int_0^L \phi_{j,x}(x)\phi_{i,x}(x)dx &= \int_0^L q(x)\phi_i(x)dx, \\ \text{or } \sum_{j=1}^n b_j C_{i,j} &= q_i \end{aligned}$$

for $i = 1, \ldots, n$.

We will show that the stiffness matrix C is tridiagonal for this example. We are solving the above system for coefficients b_j, thus we are solving $C\vec{b} = \vec{q}$ and can use iterative methods in FEM solutions too.

We can show that the stiffness matrix is tridiagonal:

$$C_{ij} = \int_0^L \phi_{j,x}(x)\phi_{i,x}(x)dx = \begin{cases} \frac{-1}{\Delta x}, & i = j-1 \\ \frac{2}{\Delta x}, & i = j \\ \frac{-1}{\Delta x}, & i = j+1 \\ 0, & \text{elsewhere} \end{cases}$$

We approximate q_i using:

$$\begin{aligned} q_i &= \int_0^L q(x)\phi_i(x)dx \approx q(x_i)\int_0^L \phi_i(x)dx \\ &= q(x_i)\left(\int_{x_{j-1}}^{x_j} \frac{1}{\Delta x}(x-x_{j-1})dx + \int_{x_j}^{x_{j+1}} 1 - \frac{1}{\Delta x}(x-x_j)dx\right) \\ &= \Delta x q(x_i) \end{aligned}$$

We can substitute the above simplifications into equation 9.4 and arrive at $C\vec{b} = \Delta x\vec{q}$ or $\frac{1}{\Delta x}C\vec{b} = \vec{q}$:

$$\begin{pmatrix} \frac{2}{\Delta x^2} & \frac{-1}{\Delta x^2} & 0 & \cdots \\ \frac{-1}{\Delta x^2} & \frac{2}{\Delta x^2} & \frac{-1}{\Delta x^2} & \ddots \\ 0 & \ddots & \ddots & \ddots \\ \vdots & \ddots & \frac{-1}{\Delta x^2} & \frac{2}{\Delta x^2} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{pmatrix}$$

Thus the matrix system to solve is the same as FD solution in this example and the solution involves inverting the stiffness matrix C:

$$\vec{b} = C^{-1}\Delta x\vec{q}$$

Iterative methods are useful in FEM too as it involves inverting large, sparse matrices.

Once $\vec{b}$ is known, the solution U to the PDE is given by:

$$U(x) \approx \sum_{j=1}^{n} b_j\phi_j(x)$$

This is a weak solution of the PDE $-U_{xx} = q$.

9.2 Comparing FEM solution to FD solution for our example

$$-U_{xx} = q, \quad 0 \le x \le L, \quad U(0) = U(L) = 0$$

9.2.1 FD solution

Discretise using $x_j = j\Delta x,\ j = 0, 1, \ldots, n+1$ where $\Delta x = \frac{L}{n+1},\ U_0 = 0 = U_{n+1}$ (using boundary conditions).

We let $U(x_j) = U_j,\ q(x_j) = q_j$. The central difference approximation to the PDE is:

$$\begin{aligned} U_{xx} &= \frac{U_{j+1} - 2U_j + U_{j-1}}{\Delta x^2} \\ \text{and} \qquad -U_{xx} &= q \quad \text{becomes} \\ -\frac{U_{j+1} + 2U_j - U_{j-1}}{\Delta x^2} &= q_j \end{aligned}$$

We solve for $U_1, \ldots, n$ since U_0 and U_{n+1} given from boundary conditions and we can rewrite in matrix form:

$$\underbrace{\begin{pmatrix} \frac{2}{\Delta x^2} & \frac{-1}{\Delta x^2} & 0 & \cdots \\ \frac{-1}{\Delta x^2} & \frac{2}{\Delta x^2} & \frac{-1}{\Delta x^2} & \ddots \\ 0 & \ddots & \ddots & \ddots \\ \vdots & \ddots & \frac{-1}{\Delta x^2} & \frac{2}{\Delta x^2} \end{pmatrix}}_{\text{same coefficient matrix for FD as FEM}} \qquad \begin{pmatrix} U_1 \\ U_2 \\ \vdots \\ U_n \end{pmatrix} + \begin{pmatrix} -\frac{U_0(=0)}{\Delta x^2} \\ 0 \\ \vdots \\ -\frac{U_{n+1}=0}{\Delta x^2} \end{pmatrix} = \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{pmatrix}$$

In this example FEM and FD methods solve the same matrix system.

9.3 2-D Finite Element Method

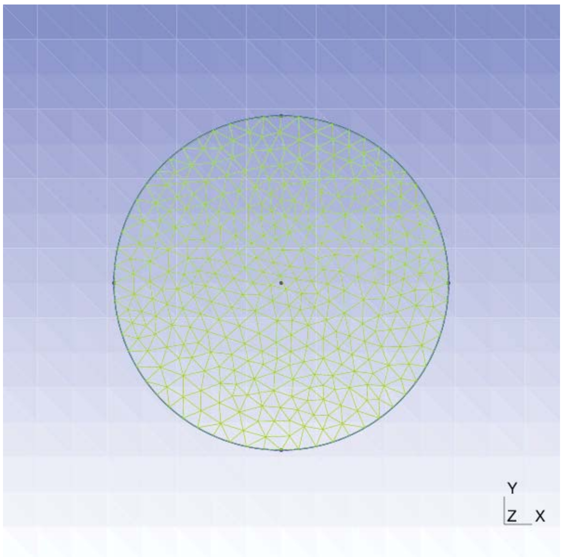


Figure 9.2: FEM mesh with triangles

We consider a triangular mesh (could also be rectangular) shown in figure 9.2 where G is the domain inside the circle and ∂G is the domain's boundary.

We are solving:

$$-\nabla^2 U = q \text{ in } G \tag{9.5}$$

with boundary conditions, $U = 0$ on ∂G.
The weak solution for U satisfies the variational form for Equation 9.5:

$$-\int\int_G \nabla^2 U(x,y)\phi(x,y)dxdy = \int\int_G q(x,y)\phi(x,y)dxdy \tag{9.6}$$

where $\phi(x,y) = 0$ on ∂G (satisfies boundary conditions).
Using Green's first identity:

$$\begin{aligned}\int\int_G (\phi\nabla^2 U)dxdy &= \underbrace{\oint_{\partial G} \phi(\nabla U \cdot \hat{n})dS}_{=0 \text{ since } \phi=0 \text{ on } \partial G} - \int\int_G (\nabla\phi \cdot \nabla U)dxdy \\ &= -\int\int_G (\nabla\phi \cdot \nabla U)dxdy\end{aligned}$$

Thus equation 9.6 becomes:

$$\int\int_G (\nabla\phi \cdot \nabla U)dxdy = \int\int_G q\phi dxdy \tag{9.7}$$

which holds $\forall \phi \in G$ where $\phi = 0$ on ∂G.
Similarly to the 1-D case we seek an approximate solution to equation 9.7 by expanding $U(x,y)$ in a basis of 2-D *"hat"* functions:

$$U(x,y) \approx U_h(x,y) = \sum_{j=1}^{n} b_j\phi_j(x,y)$$

where $U_h(x_i, y_i) = b_i$ and $U_h = 0$ on ∂G.

9.3.1 2-D "hat functions"

The 2-D hat functions satisfy $\phi_j(x_j, y_j) = 1, \quad \phi_j(x_i, y_l) = 0$ if $i \neq j$ and $j \neq l$ at all other vertices. The 2D hat function is plotted in figure 9.3.

We require that equation 9.7 holds for all $\phi(x,y)$ and solve for $\phi = \phi_1, \phi_2, \ldots, \phi_n$:

$$\int\int_G \nabla U_h \cdot \nabla\phi_i dxdy = \int\int_G q\phi_i dxdy, \text{ for } i = 1, \ldots, n \tag{9.8}$$

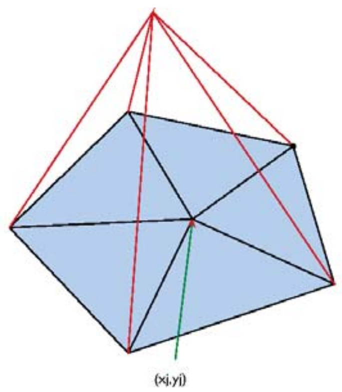


Figure 9.3: 2D hat function ($\phi_j(x_j, y_j) = 1, \quad \phi_j(x_i, y_l) = 0$ if $i \neq j$ and $j \neq l$)

$$LHS = \int\int_G \nabla U_h \cdot \nabla \phi_i dxdy = \sum_{j=1}^{n} C_{i,j} b_j$$

where $C_{i,j} = \int\int_G \nabla \phi_j \cdot \nabla \phi_i dxdy$ is called the "stiffness matrix".
Equation 9.8 becomes:

$$\sum_{j=1}^{n} C_{i,j} b_j = q_i, \text{ for } i = 1, \ldots, n \text{ where } q_i = \int\int_G q \phi_i dxdy.$$

If C is symmetric and positive definite then the system has a unique solution.

9.3.2 Example: 2-D Finite Element Method using eScript for elastic wave propagation from a point source.

- eScript is a general PDE solver which implements the finite element method written in python
  (see https://launchpad.net/escript-finley)

- eScript can be applied to any problem of the form:

$$-(A_{jl} a_{,l} + B_j a)_{,j} + C_l a_{,l} + Da = -X_{j,j} + Y$$

  where a is the *scalar* we are solving for in this example.
  (eScript can also solve for a *vector* $\vec{a}$)

We are using *Einstein notation* and according to this convention if an index appears *twice* in a single term it implies we are summing over all possible values:

$$a_i f_{,ii} = a_i \frac{\partial^2 f_i}{\partial x_i^2} = a_1 \frac{\partial^2 f_1}{\partial x_1^2} + a_2 \frac{\partial^2 f_2}{\partial x_2}$$

We will see that the FEM takes care of *spatial* derivative in the problem below. However we still need to approximate *time* derivatives.

We want to solve the 2-D wave equation for a point source:

$$\Psi_{tt} = V_p^2(\Psi_{xx} + \Psi_{yy}) + F_{\mathrm{PS}}$$

where p is the wave speed, Ψ is the wave-field and F_{PS} is the force due to the point source.
In eScript this becomes:

$$\begin{aligned} Da &= -X_{j,j} + Y \\ \text{where } a &= \Psi_{tt} \\ D &= 1 \\ X_j &= -V_p^2 \Psi_{,j} \\ Y &= F_{\mathrm{PS}} \end{aligned}$$

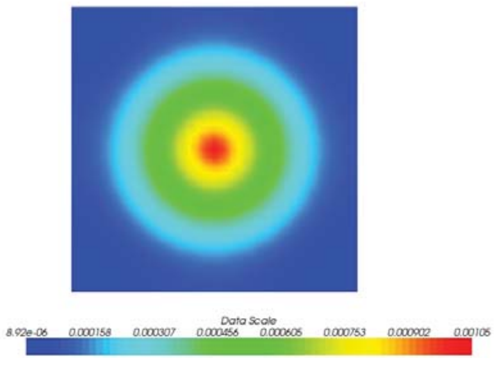


Figure 9.4: Plot of Euclidean normal of the displacement at $t > 0$ for a point source using eScript.

We solve for a^k at each time step t^k. Once a^k is known we use it to calculate the solution at the next time step, Ψ^{k+1} using the central difference formula:

$$a^k = \frac{\partial^2 \Psi^k}{\partial t^2} \approx \frac{\Psi^{k+1} - 2\Psi^k + \Psi^{k-1}}{\Delta t^2}$$

or

$$\Psi^{k+1} = 2\Psi^k - \Psi^{k-1} - \Delta t^2 a^k$$

The eScript python code is **2Dpointsource.py** and the output from this code is shown in figure 9.4.